



PushBot—Program It

The PushBot has a LOT to do! Because of this, I want to be absolutely certain that my program works, and I'll be testing it often. To make it easier on myself, I'm going to break down the program into three sections. The first section will cover getting the bot to the proper location in front of the first figurine. The second section will cover the programming blocks used to put the first three figurines into their proper locations. And the final section will be devoted to pushing the last figurine up the ramp and onto the Mayan sarcophagus.

If you've built your own version of the PushBot, your program will most likely not match mine exactly. It really depends on how you've built your bot to complete the challenge.

What I'll be doing in the next three sections is walking you through the programming of this last bot so the team can finally enter King Ixtua's burial chamber and see what it contains. So, without any further delay, let's get this bot programmed and ready to go.

Getting the PushBot into Position

In the Lego Mindstorms NXT software, I'll type **PushBot** in the blank text field labeled **Start New Program** and then click **Go** (see Figure 20-1).

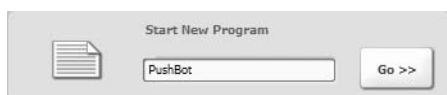


Figure 20-1. *The beginning of the program for the PushBot*

Note If the RoboCenter area is visible on the far right of your screen, click the small red X in the upper-right corner of the software, and it will free up more visible workspace on your screen.

Once again, I am using the Touch Sensor as a **Start** button for the PushBot. You can choose to use one of the Brick's buttons, but I like the simplicity of running the program and having it wait until I press the Touch Sensor. I've placed the Touch Sensor in an easy-to-reach place on top of the Brick.

The first block I've dropped in my program is a LOOP block (see Figure 20-2) that will wait for the Touch Sensor to trigger.

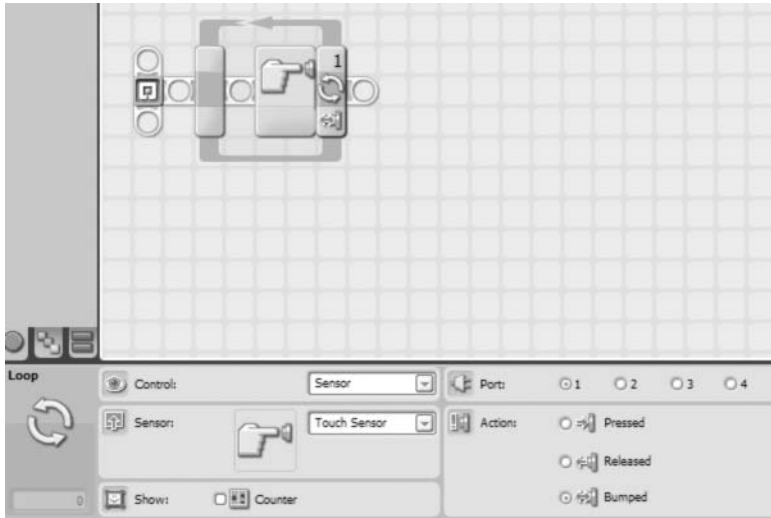


Figure 20-2. This *LOOP* block will break when the *Touch Sensor* is triggered.

The first task my bot must perform is “Move from pedestal to first figurine without touching it.” I’m going to take advantage of the fact that I can place my PushBot initially on the floor and point it directly at the first figurine. After that, I have a couple of different methods available for getting the bot into position.

I know roughly the distance between the pedestal and the first figurine, so as long as I program the bot to stop before it touches the figurine, I should be safe. If you look back to Figure 17-2, you see that the estimated distance is 7 feet. I have measured my PushBot and it is just a little over 18 inches long (1.5 feet). Take a look at Figure 20-3. If I place my bot as shown and point it toward the first figurine, it will need to move forward 5.5 feet and then stop to avoid touching the figurine. So, for the first method, I could program a *MOVE* block with the correct number of rotations or degrees to move it 5.5 feet.

But I want to use the second method that relies on the Ultrasonic Sensor. Why do I want to do this? Well, if my measurements are off by just an inch or two, I might accidentally bump one of the figurines. I’ve noticed that when the batteries get low in my bot, sometimes strange things can happen with the motors. What I’d prefer is to use the Ultrasonic Sensor to detect the figurine and stop the bot a safe distance away. And to accomplish this, I’ll simply program my bot to move slowly toward the figurine, constantly checking to see whether it detects an obstacle (figurine).

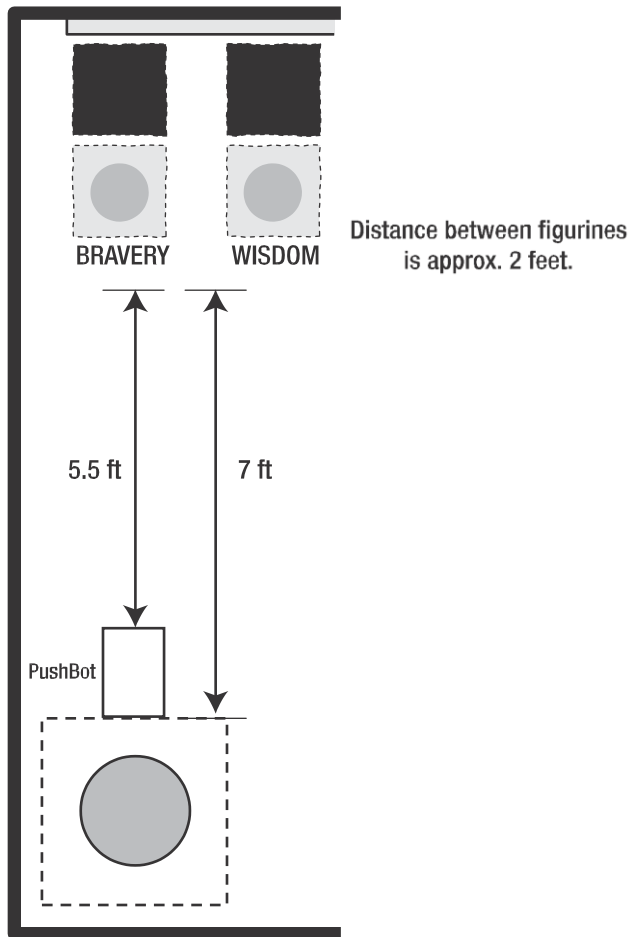


Figure 20-3. *The PushBot's initial movement will be 5.5 feet forward.*

In Figure 20-4, you'll see that I've dropped in a LOOP block that is configured to break when the Ultrasonic Sensor is triggered. Since I haven't tested yet, I'll set the Ultrasonic Sensor to be triggered when it detects an obstacle less than 12 inches away. Remember, this might change once I test the sensitivity of the Ultrasonic Sensor.

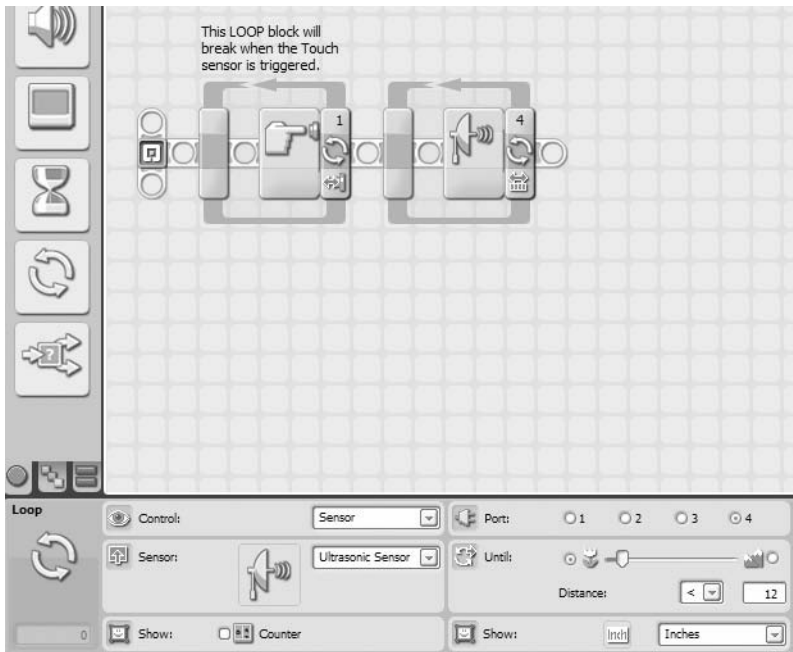


Figure 20-4. This *LOOP* block will break when the Ultrasonic Sensor is triggered.

Before I perform an initial test of my PushBot, I'll place a **MOVE** block inside the **LOOP** and tell it to spin motors B and C. I can choose to set the **Duration** of the motors either to **Unlimited** or for a specific number of degrees or rotations. I will test both methods to determine which is the safest way to approach the figurine. You can see the **MOVE** block in Figure 20-5, and this block is initially configured to spin for one rotation. I am doing this because I want the bot to move forward, the Ultrasonic Sensor to test for distance, and then the bot to move forward one rotation again; this will continue until the Ultrasonic Sensor is triggered. (I'll also test it with the motors spinning constantly until the Ultrasonic Sensor is triggered.)

I place a bottle of water about 9 feet away from the PushBot. I push the Touch Sensor (**Start** button) and off it goes. Results? When I configure the **MOVE** block **Duration** to **Unlimited**, the bot stops much nearer to the bottle than when I configure it for a **Duration** of one rotation. I also am forced to increase the sensitivity of the Ultrasonic Sensor to 15 inches or less. After testing again, I find these settings are acceptable, and the PushBot is about 12 to 15 inches from the bottle.

Now that the bottle is found, I need to configure the PushBot to push the bottle. My PushBot design requires the bot to spin around and move toward the bottle until I tell it to stop. It will then close the cage mechanism I designed and hold the bottle as the bot pushes it forward. The bot will stop when the Light Sensor is triggered by the black pressure plate (obsidian rock). Let's take all of this one step at a time.

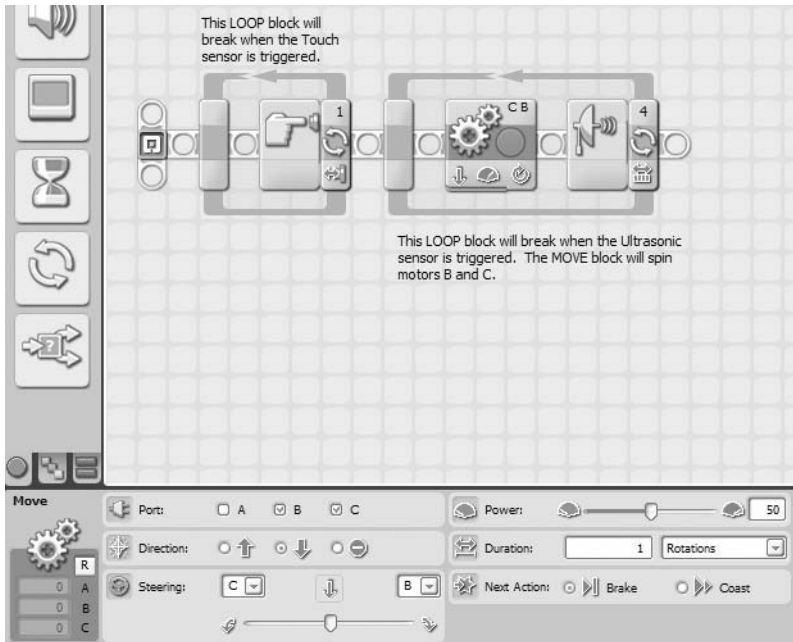


Figure 20-5. A MOVE block for the PushBot

First, I need the bot to turn 180 degrees so the cage is facing the bottle. How do I do this? Simple. I first back the bot up to give it room to turn—one rotation of motors B and C should do it. I use a MOVE block to do this, as shown in Figure 20-6.

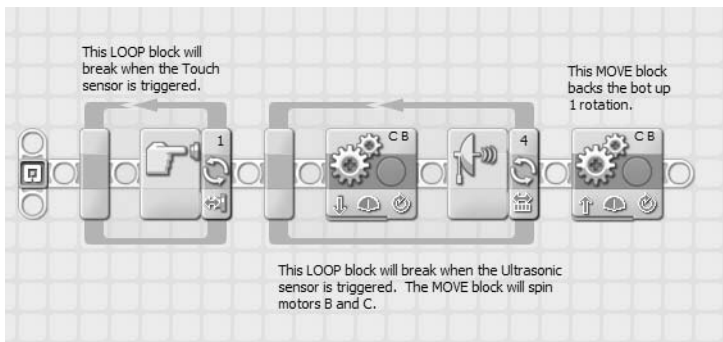


Figure 20-6. The PushBot will first back up about 6 inches.

Next, I will turn ONLY motor B so the bot turns 90 degrees to the left. I will follow this with a MOVE block that turns ONLY motor C so the bot turns 90 degrees to the right. At this point, the PushBot will be facing the reverse direction. This is demonstrated in Figure 20-7.

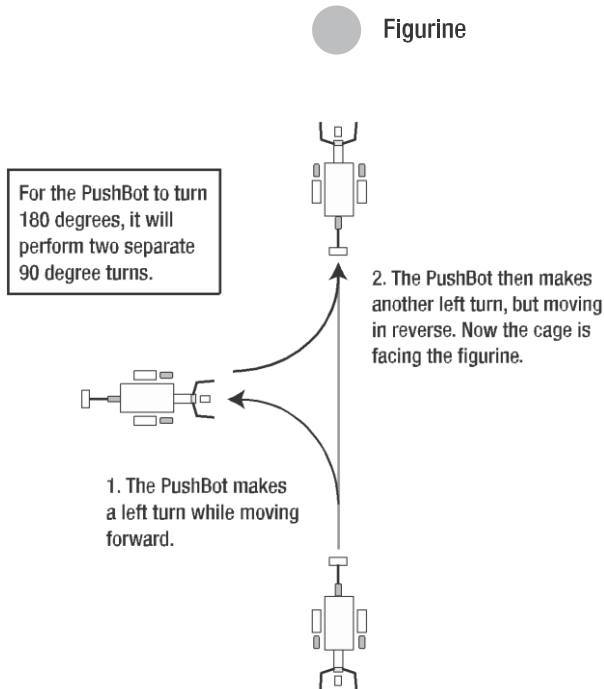


Figure 20-7. *The PushBot can turn 180 degrees by a combination of MOVE blocks.*

I used the method described in Chapter 4 for obtaining the correct number of degrees for motors B and C to turn. For my PushBot, the right turn (motor B) made in Figure 20-7 requires 508 degrees. For the left turn (motor C), the motor is configured for -513 degrees (the negative sign simply tells me to configure motor C for 513 degrees but to also change the spin direction). You cannot use a negative number as a duration, so you must enter the same value (513) but select the opposite spin direction. Figure 20-8 shows the MOVE block for the right turn.

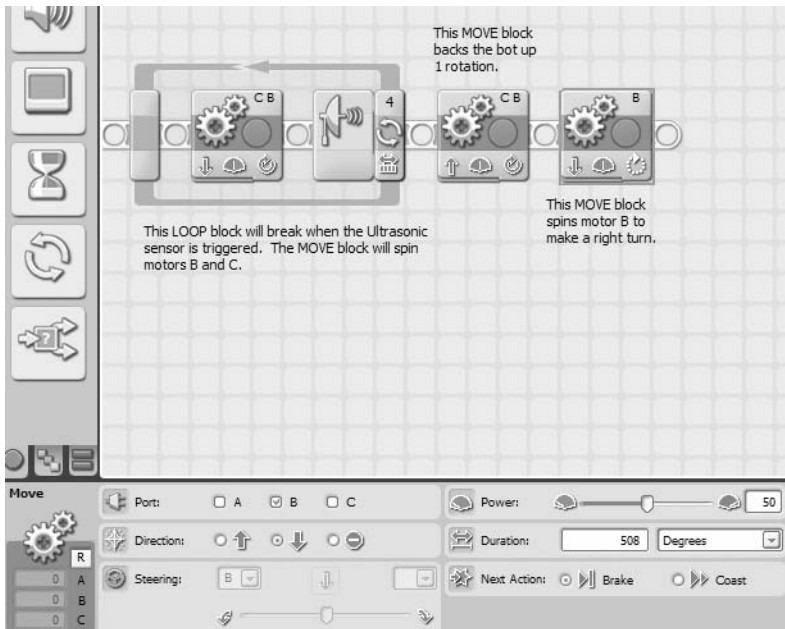


Figure 20-8. This MOVE block allows the PushBot to make a right turn.

I follow this with another MOVE block for the left turn in Figure 20-9.

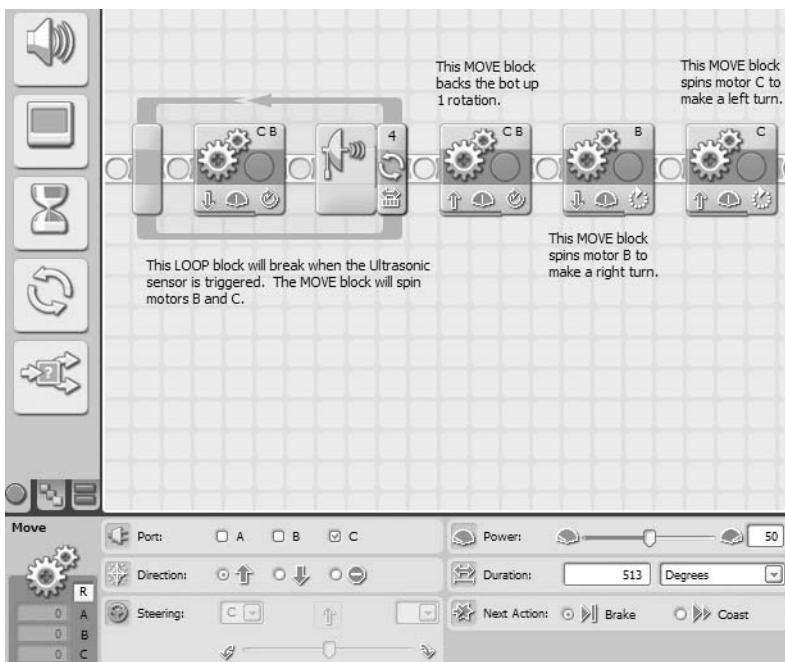


Figure 20-9. And this MOVE block allows the PushBot to turn left.

I save and upload the program to my PushBot for testing. After testing, the PushBot is lined up properly with the bottle.

Now, let's pause here for just a minute and look at where my PushBot is located. Currently, the bot has spun around and is facing the first figurine with the cage in the opened position. What happens next? Let me break it down in a small list:

1. Approach figurine 1.
2. Close the cage around figurine 1.
3. Push figurine 1 forward onto pressure plate and stop.
4. Open cage and reverse to starting position.
5. Turn right, move forward a short distance, and then turn left.
6. PushBot is in front of figurine 2.

Do you see the pattern? It will do the exact same steps for figurine 2 and then end up facing figurine 3. It will perform the steps for figurine 3 with the only difference being that after it opens the cage (step 4) and reverses direction, it will need to be directed to the figurine near the ramp.

Because of these repeated steps, I'm going to use a LOOP block that will allow me to perform the preceding steps three times, once for each of the first three figurines.

Positioning Three Figurines

Before my bot locates and begins pushing the final figurine up the ramp, I'll need it to be stopped in front of figurine 3. Knowing this information will help me to determine what I should place inside the LOOP block I'm going to add for the actions needed to push figurines 1, 2, and 3 into position.

But first, I'll drop in the LOOP block (see Figure 20-10) and configure it to execute any blocks placed inside it one time. (I'll later change this to three times after testing so it will perform the same actions for three bottles.)

Now, the first thing I need the PushBot to do is move slowly toward the figurine and stop when the Sound Sensor detects my voice. I'll use another LOOP block and configure it to break when the Sound Sensor is triggered (see Figure 20-11). During testing I'll fine-tune the sensitivity of the Sound Sensor, but for now I'm setting it to detect any sound greater than 50.

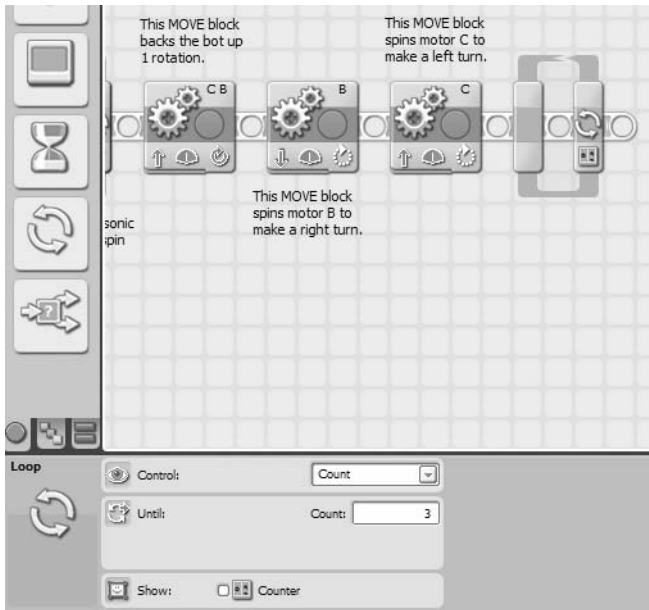


Figure 20-10. The *LOOP* block for positioning the first three figurines

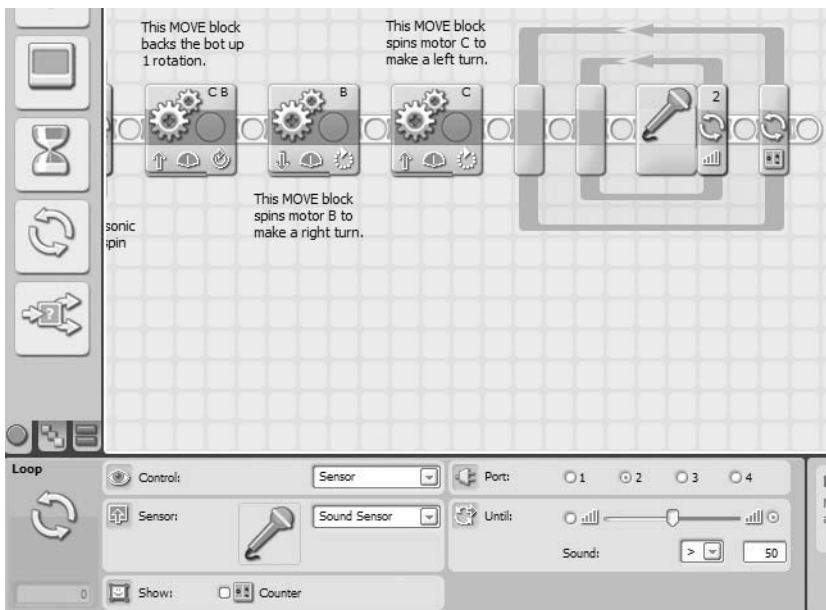


Figure 20-11. This inner *LOOP* block will break when the *Sound Sensor* is triggered.

I'll add in a *MOVE* block that will slowly move the bot toward the bottle. Figure 20-12 shows that I've configured it to spin motors B and C .25 rotation (a quarter rotation).

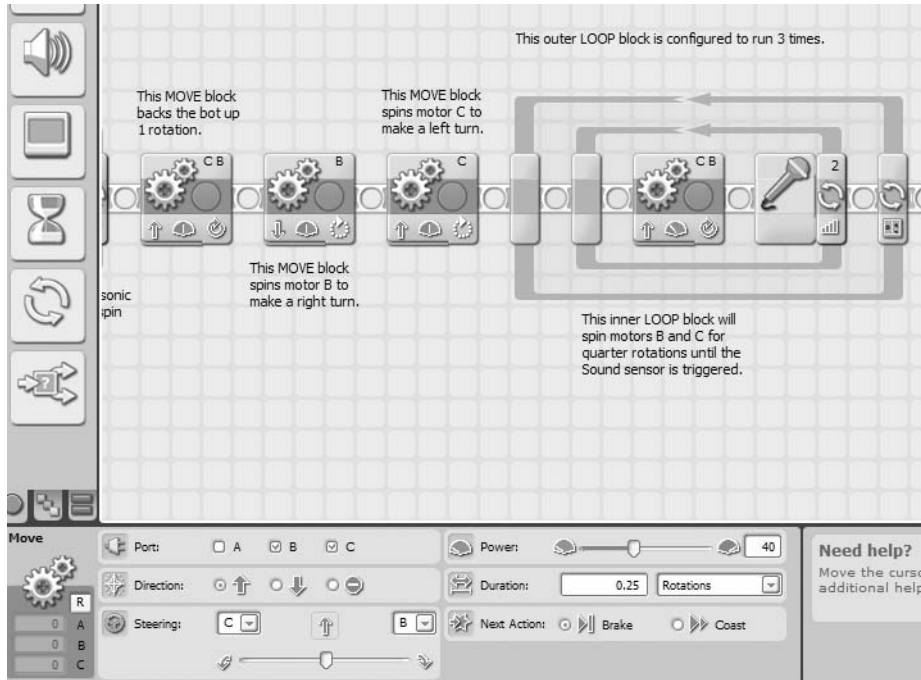


Figure 20-12. The MOVE block will spin motors B and C in quarter rotations.

Now it's time to upload the program and test it. I had to change the sensitivity of the Sound Sensor to a value of 70; normal noises around my house were setting off the sensor.

The bot did a perfect turn and approached the bottle slowly until the Sound Sensor triggered. Now I want the cage to close and hold the bottle. This is done with a MOVE block as shown in Figure 20-13.

I configured the MOVE block to spin motor A -65 degrees. In Figure 20-13, I enter a value of **65** for the number of **Degrees** and I also configure motor A to spin in reverse (to take into account the negative value of the rotation). This was the proper number of degrees required to move the cage from an open position to the closed position. I obtain this value using the **View** option on the Brick; refer to Chapter 4 for a review on how to obtain this reading.

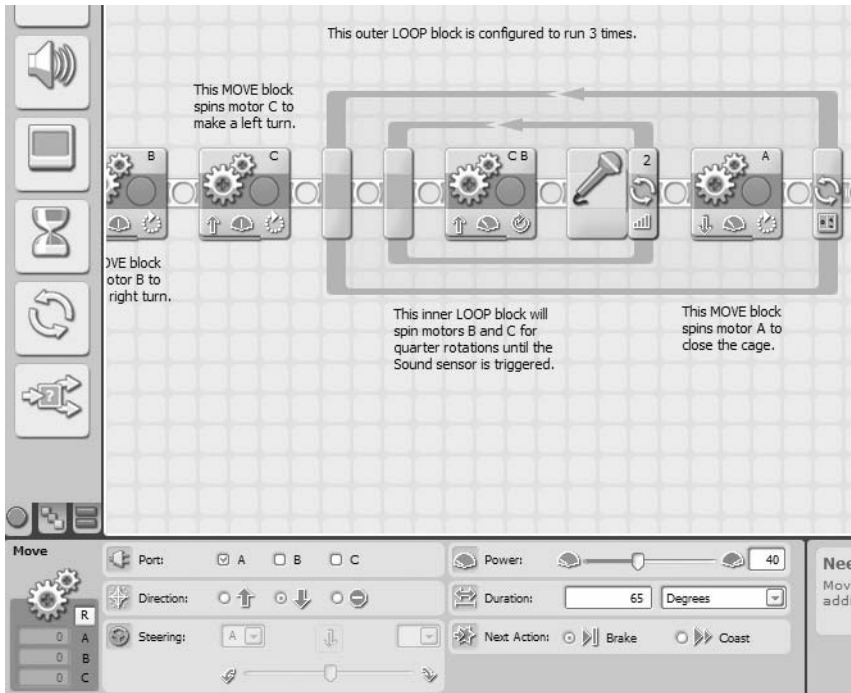


Figure 20-13. *The MOVE block stops motor A to save battery power.*

After the cage closes, I've added in another MOVE block that turns off the power to motor A (see Figure 20-14), saving battery power.

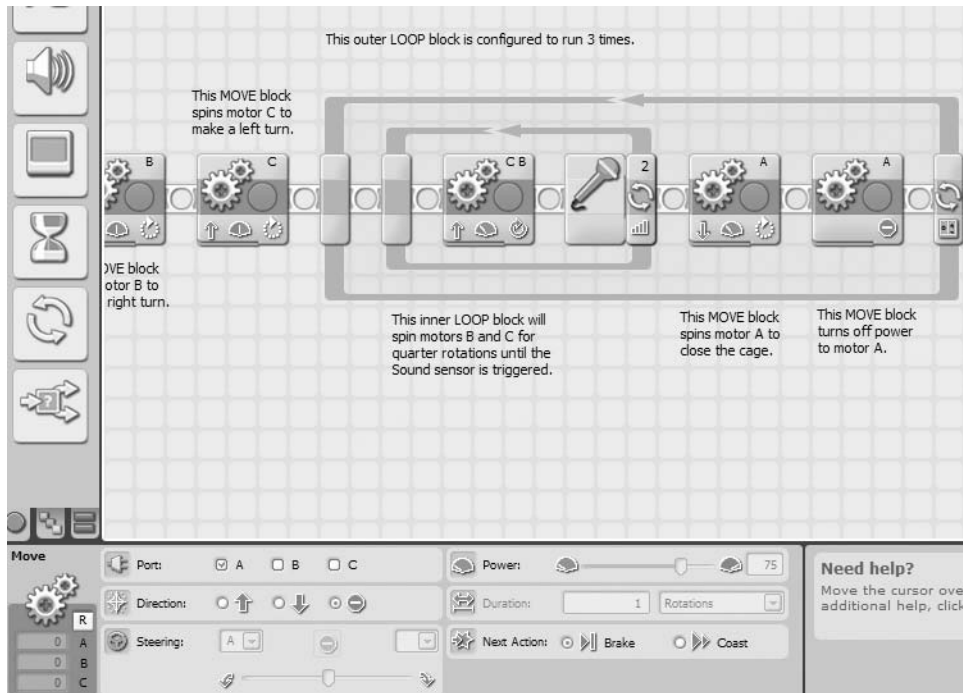


Figure 20-14. This MOVE block turns off power to motor A.

Now that the cage is closed, I need the bot to slowly begin pushing the figurine (bottle) until it detects the black obsidian rock floor. For testing, I've taped a black square of paper to the floor behind the bottle. I'll use a LOOP block that will break when the Light Sensor is triggered (see Figure 20-15). Once again, I test the Light Sensor on the black paper to obtain the correct setting (using the instructions in Chapter 4 to use the **View** option on the Brick). The black paper gives me a value of 3; the floor gives me a value of 12. I will set the Light Sensor to trigger if the value drops below 5. Your values may differ, so testing is absolutely required.

I need to drop in a MOVE block to perform the pushing action (see Figure 20-16). I'll have it push forward 3.5 inches at a time (half a rotation).

Note Notice that motors B and C are frequently changing spin direction. In some instances you might think your robot will move forward, but it moves backward. When I had the PushBot spin around 180 degrees, everything changed! Now when I want motors B and C to spin forward, I have to configure their direction differently. It can get confusing, but that's why we test.

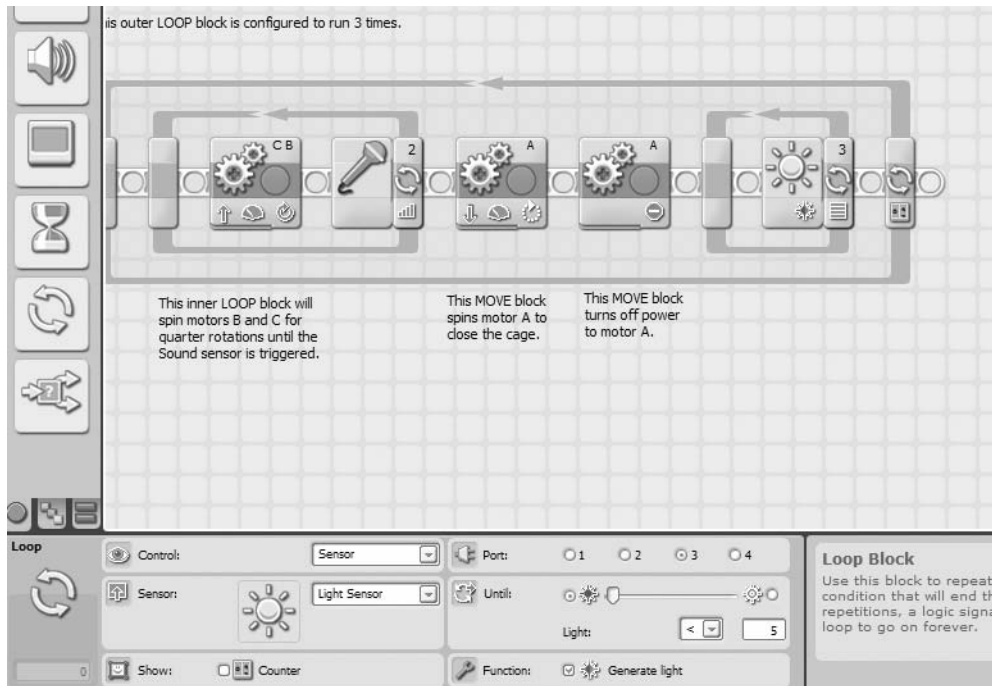


Figure 20-15. The LOOP block will break when the Light Sensor is triggered.

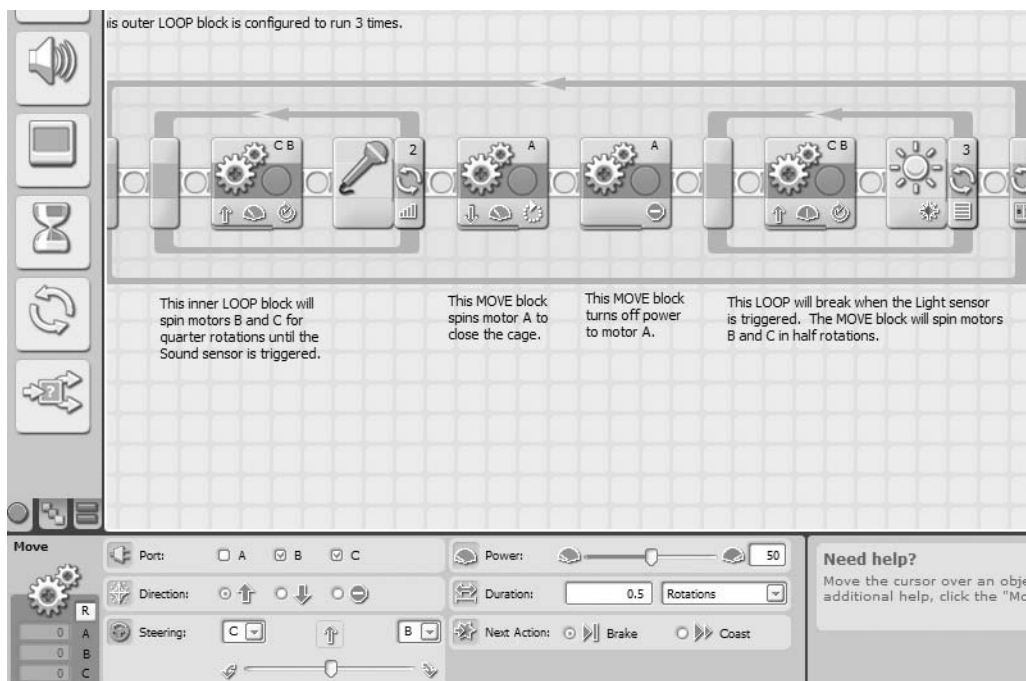


Figure 20-16. The MOVE block pushes the figurine forward in half rotations.

Now it's time to upload and test. If everything works as planned, my bot should approach the bottle, stop when the bottle is detected, back up a bit and then spin around, approach the bottle until I tell it to stop, close the cage, and then push the bottle forward until the Light Sensor is triggered.

Does it work? Not perfectly. That's the great thing about testing frequently. I don't reach the end of my program only to find that I made a mistake very early in the program. The problem is easy to fix. Motor A spins 65 degrees, but sometimes it doesn't spin all the way closed. The motor stalls and the program doesn't continue until motor A finishes its movement and closes the cage completely. My fix is to simply have motor A spin for 1 second. I'll leave the MOVE block in that stops motor A, just in case I later decide to switch back to closing it a specified number of degrees or rotations. Other than that, the PushBot works well.

It pushes the bottle onto the black paper (pressure plate) and stops. Now I just need to get it to open the cage, back up a reasonable distance, and perform a couple of movements to put it in front of figurine 2.

First, I'll open the cage by placing a MOVE block (see Figure 20-17).

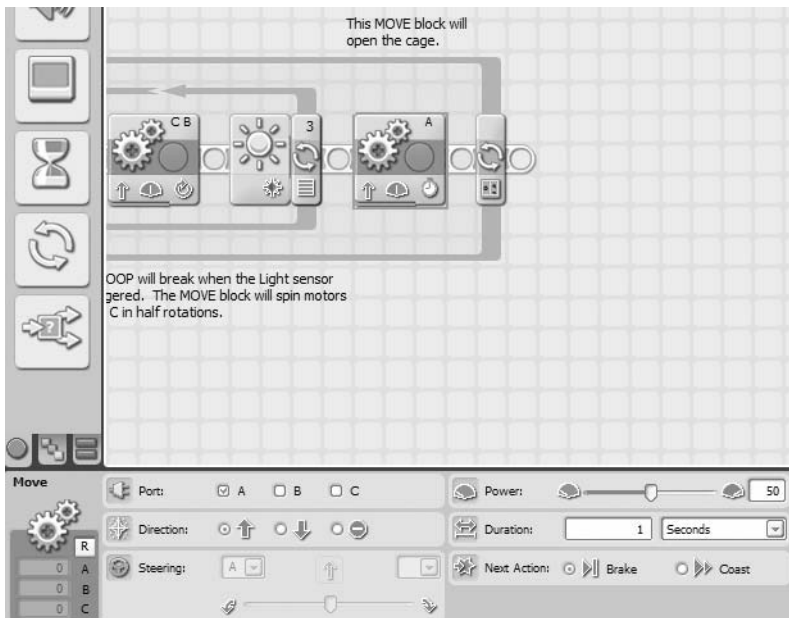


Figure 20-17. This MOVE block opens the cage.

Next, I'll get the PushBot to reverse direction a reasonable distance. Referring back to Figure 17-2, I think a safe distance for the bot to pull back would be about 3 feet. I'll test it and tweak that value if I find the bot isn't pulling back far enough. Figure 20-18 shows the MOVE block I've added. (To configure the distance, I've converted 3 feet to 36 inches. I divide that by 7 inches—the circumference of the tires—and I get a value of approximately 5 rotations.)

The last few items I need to take care of will get the PushBot in front of the next figurine. The bot needs to turn right, move forward a small distance (about 2 feet), and then turn left. This will position it facing the next figurine.

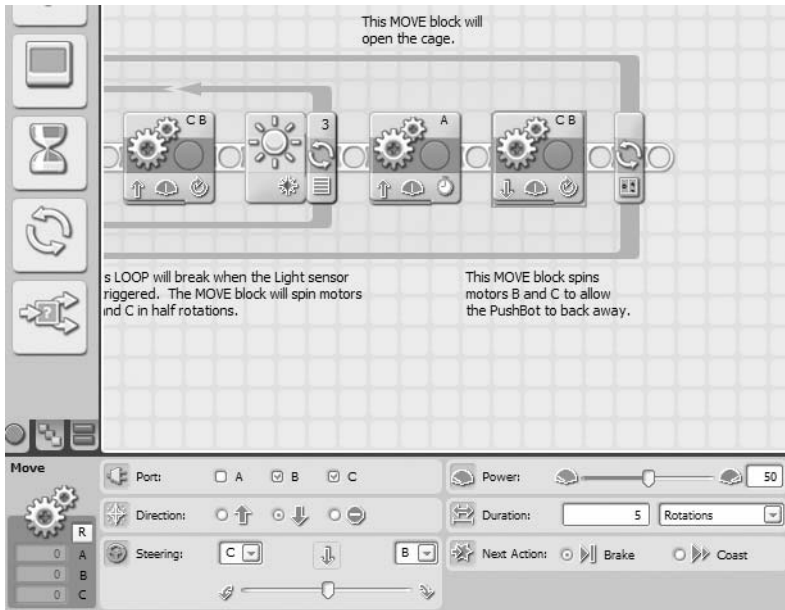


Figure 20-18. This MOVE block will allow the PushBot to move away from the figurine.

First, I'll have the bot make a right turn (see Figure 20-19).

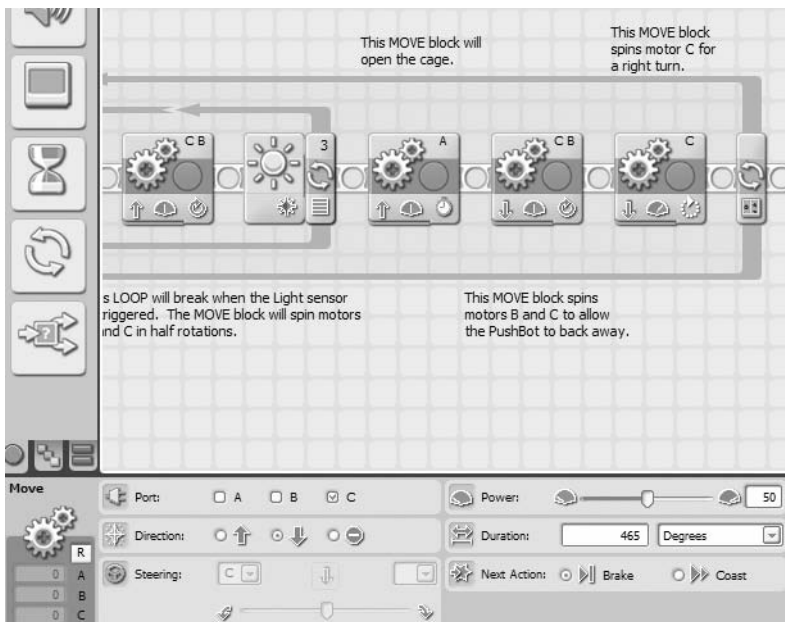


Figure 20-19. This MOVE block allows the PushBot to turn right.

Next, Figure 20-20 shows the MOVE block needed so the PushBot moves forward 2 feet. Keep in mind that this is one of those configurations that might need to be changed; you won't know until you test your bot with three figurines properly placed.

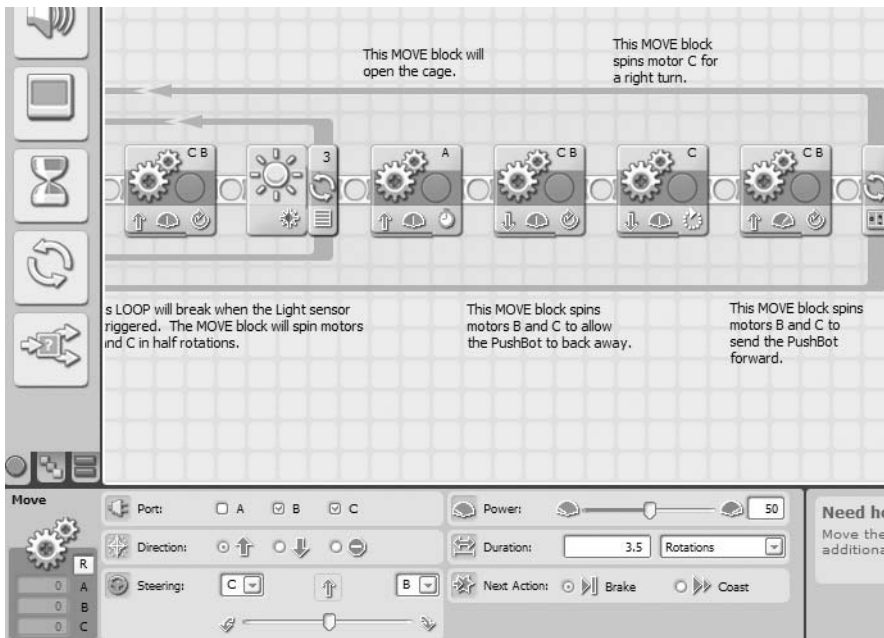


Figure 20-20. This MOVE block sends the PushBot forward approximately 2 feet.

And, finally, I need the bot to spin to the left (see Figure 20-21).

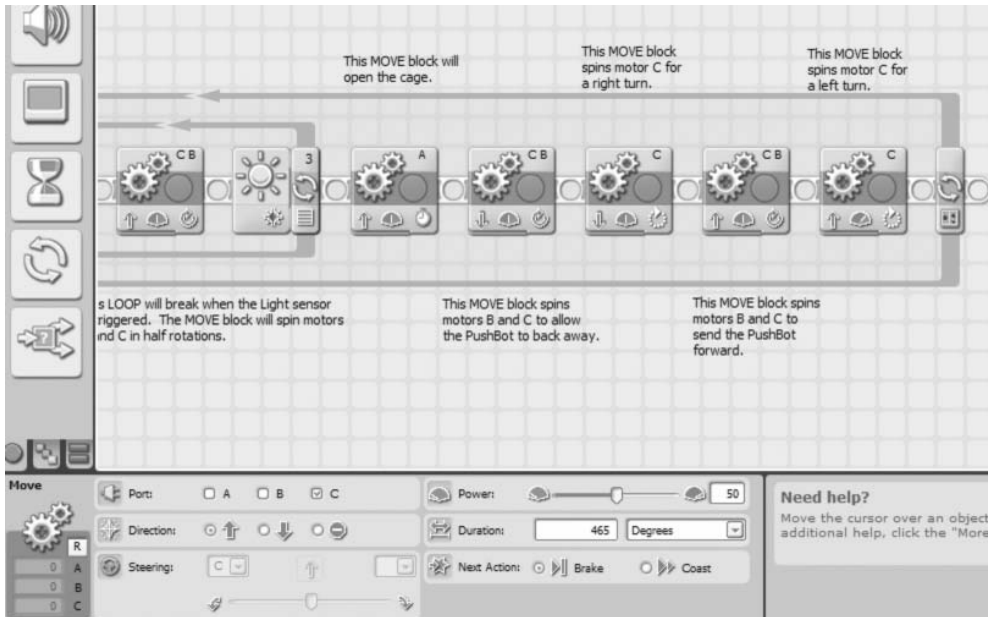


Figure 20-21. This *MOVE* block allows the *PushBot* to turn left.

Time to test; let me describe my testing environment. I place three water bottles 4 feet from a wall in my living room. Each water bottle is 2 feet apart as well. I then place three squares of black paper about 4 inches behind each bottle (the paper is taped down so it can't be pushed along with the bottle). I then place my PushBot about 9 feet away from the first bottle, pointing directly at the bottle with the Ultrasonic Sensor and with the cage fully open. And here's what happens.

I have success on the first figurine, but overshoot the second figurine by about 6 inches. I fix this by reducing the number of durations motors B and C spin (in Figure 20-20) from 5 to 4. I also have a problem with the cage closing properly but opening too wide, and the arms rub the tires. I reduce the time for the cage to open from 1 second to .5 seconds (half a second). On the second test, I have perfect success—all three figurines are located and pushed to their proper locations without tipping over.

At this point, I'd like to remind you that your settings will probably be different. Motors and sensors all have varying sensitivity, so testing is an absolute requirement at this point. Don't stress about this part of building robots; this is how you improve your skills and gain insight into the way robots work in different environments.

Figure 20-22 shows the approximate location of the PushBot at this point. I've got to get it moving so it can push the fourth figurine up the ramp and onto the sarcophagus.

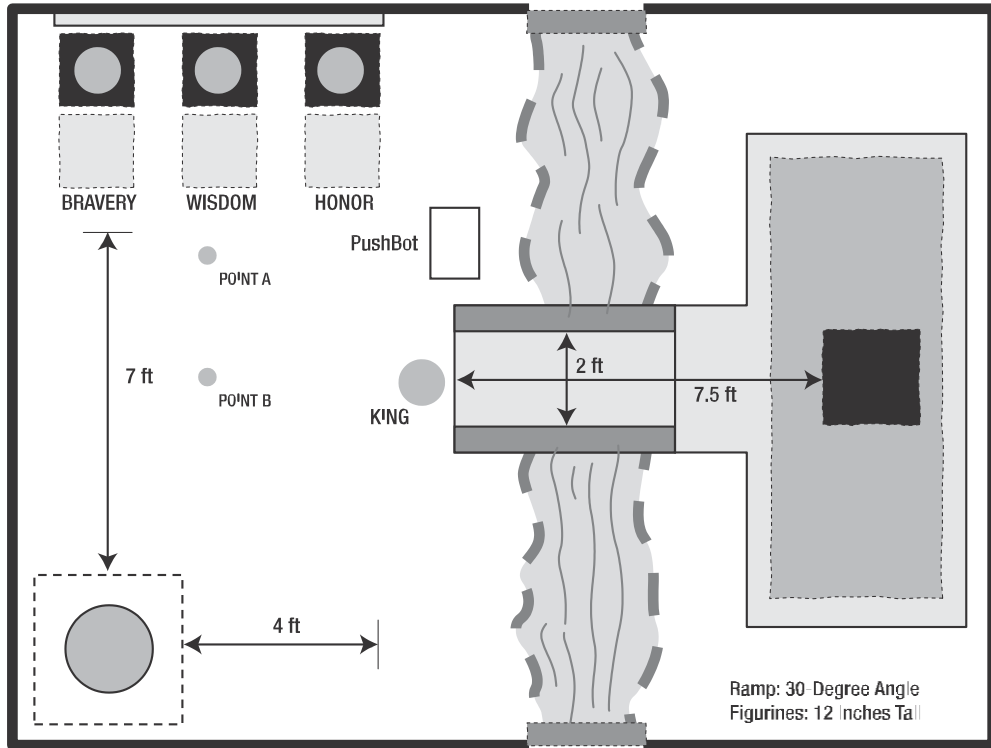


Figure 20-22. *The PushBot should be located here.*

The Final Figurine

Now to what I consider the easy part. Okay, maybe not easy, but definitely easier than trying to push three figurines onto pressure plates. I have some rough distances to work with from Figure 20-22, so I'll use these measurements, along with the Sound Sensor, to help me position the PushBot for its final task.

First, I've got to reorient the bot. Since it is pointing away from the fourth figurine, I'm first going to have the bot spin to the right and then move to Point A in Figure 20-22. The cage will be pointing in the direction of the sarcophagus when it reaches Point A. I'll drop a MOVE block in for the turn to the right (see Figure 20-23).

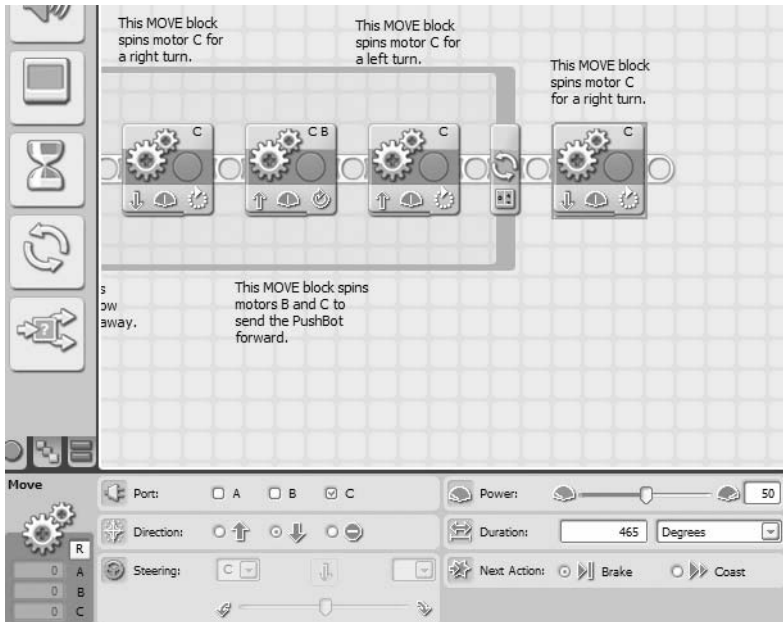


Figure 20-23. This *MOVE* block will spin the bot to the right in a clockwise direction.

Next I'll add another MOVE block that will force the bot to move to Point A—approximately 4 feet (see Figure 20-24).

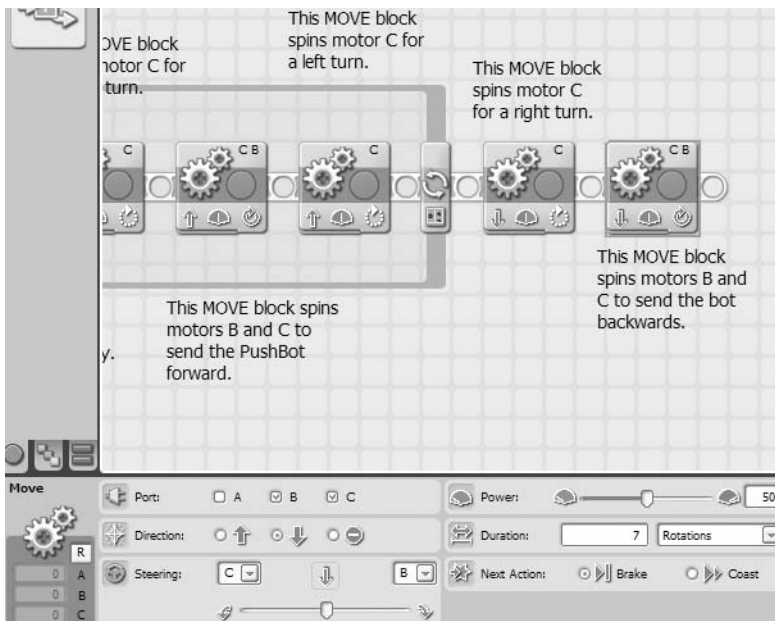


Figure 20-24. This *MOVE* block moves the bot backward to Point A.

My bot moves back seven rotations (approximately 4 feet) and stops. It's now at Point A and I'm ready to get it to point B. It's important for me to have the bot centered with the figurine and the ramp. To do this, I'll perform plenty of tests using the Sound Sensor to get it to stop and turn at precisely the right point where I want it.

I drop in another MOVE block (see Figure 20-25) to get the bot to turn right (the bot's cage will be facing away from the first three figurines).

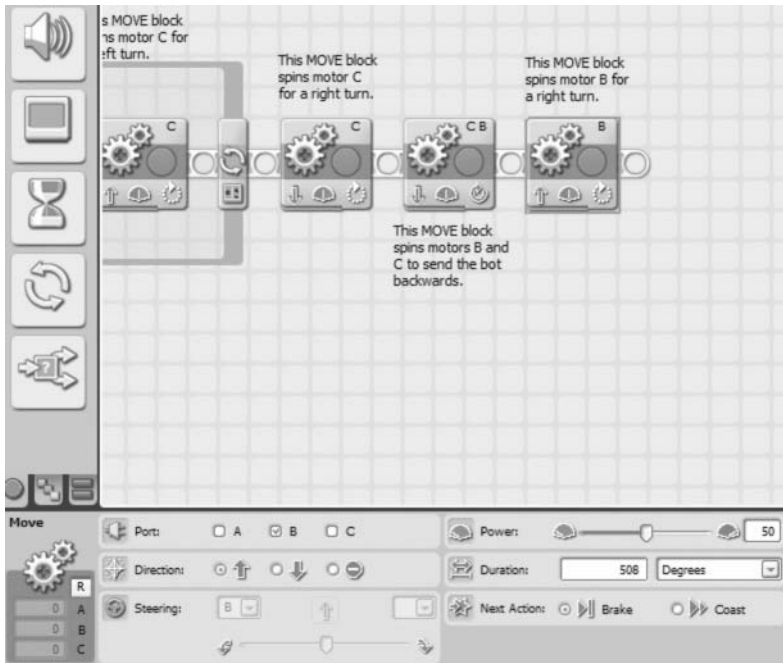


Figure 20-25. *This MOVE block turns the PushBot toward Point B.*

And now I want the bot to slowly move down to Point B. I say slowly because I'm going to watch it and trigger the Sound Sensor to stop the bot when it reaches the center point. Again, this will take some practice to determine exactly where the bot should be when I yell "Stop!" I'll add a LOOP block as shown in Figure 20-26 that will break when the Sound Sensor is triggered.

I add in the MOVE block that will slowly spin motors B and C and a WAIT block configured for 2 seconds to give me time to watch as it approaches Point B (see Figure 20-27).

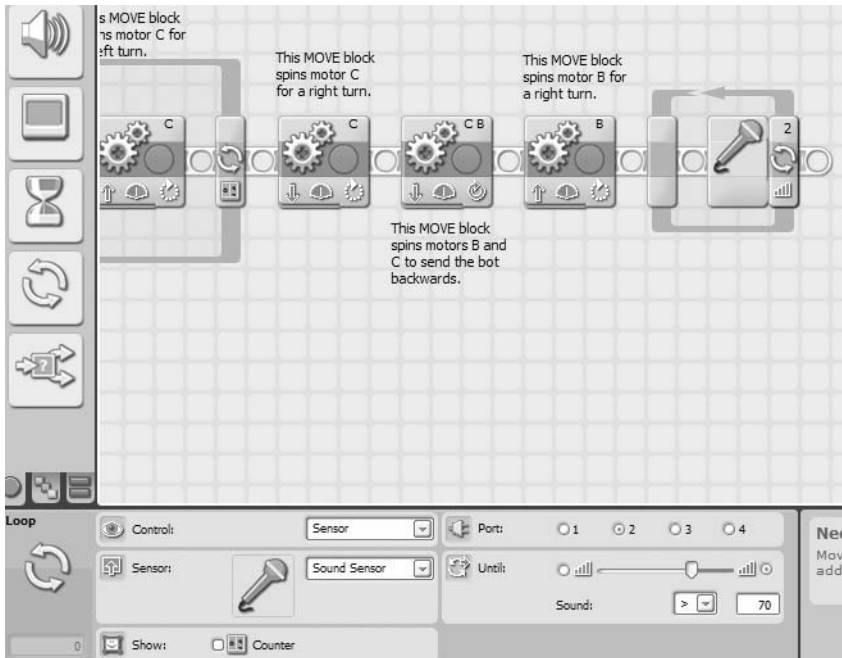


Figure 20-26. This LOOP block will break when the Sound Sensor is triggered.

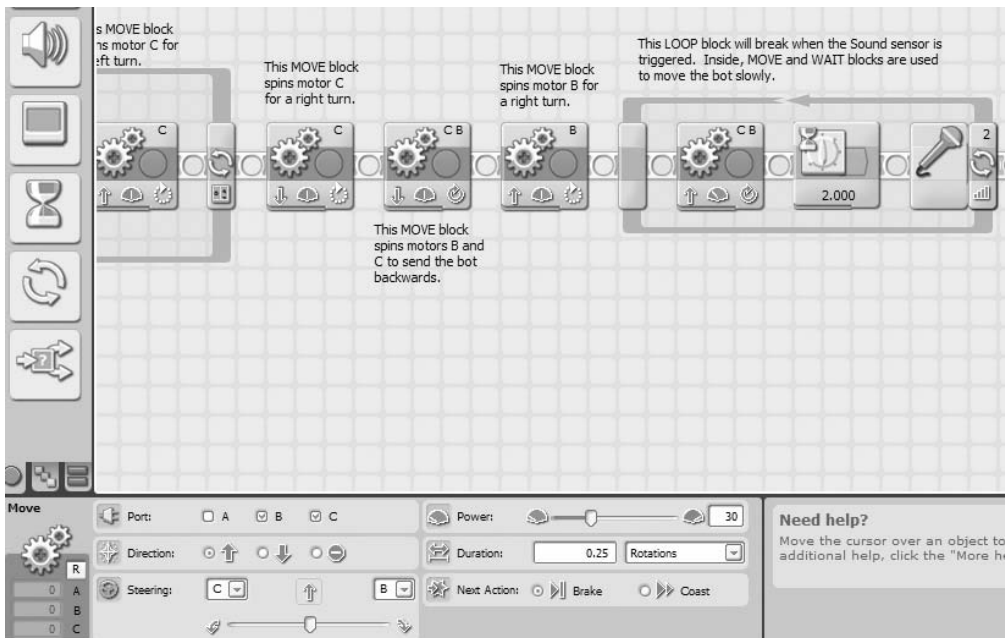


Figure 20-27. MOVE and WAIT blocks are added to move the bot to Point B.

Once it reaches Point B, it needs to make a left turn (cage facing figurine 4). I add in a MOVE block for this left turn (see Figure 20-28).

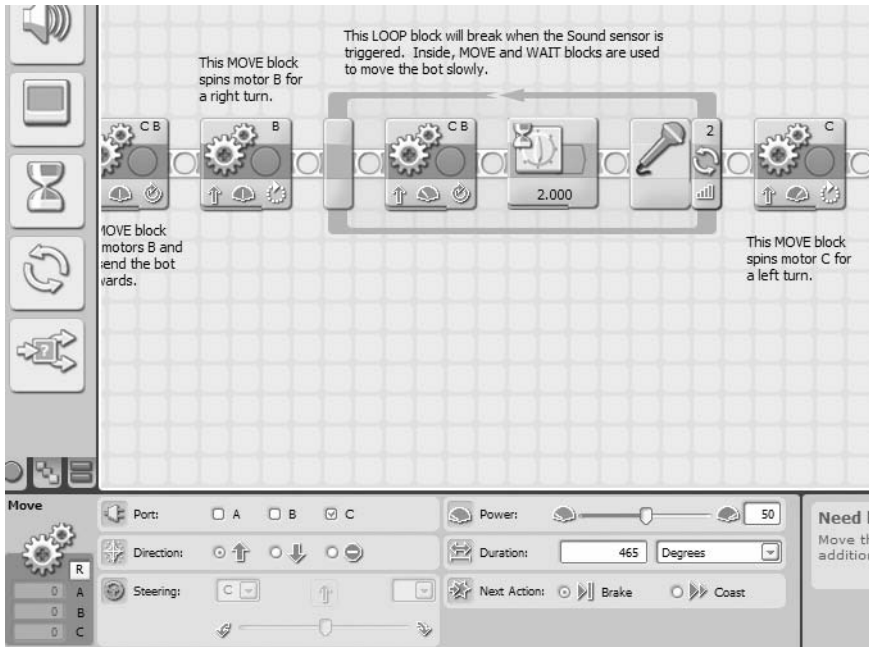


Figure 20-28. This MOVE block turns the bot to the left with the cage facing the figurine.

And now I need the bot to move (slowly) toward the fourth figurine. Once again, I'll use the Sound Sensor to tell the bot when to stop and close the cage. I add in a LOOP that is configured to break when the Sound Sensor is triggered (see Figure 20-29).

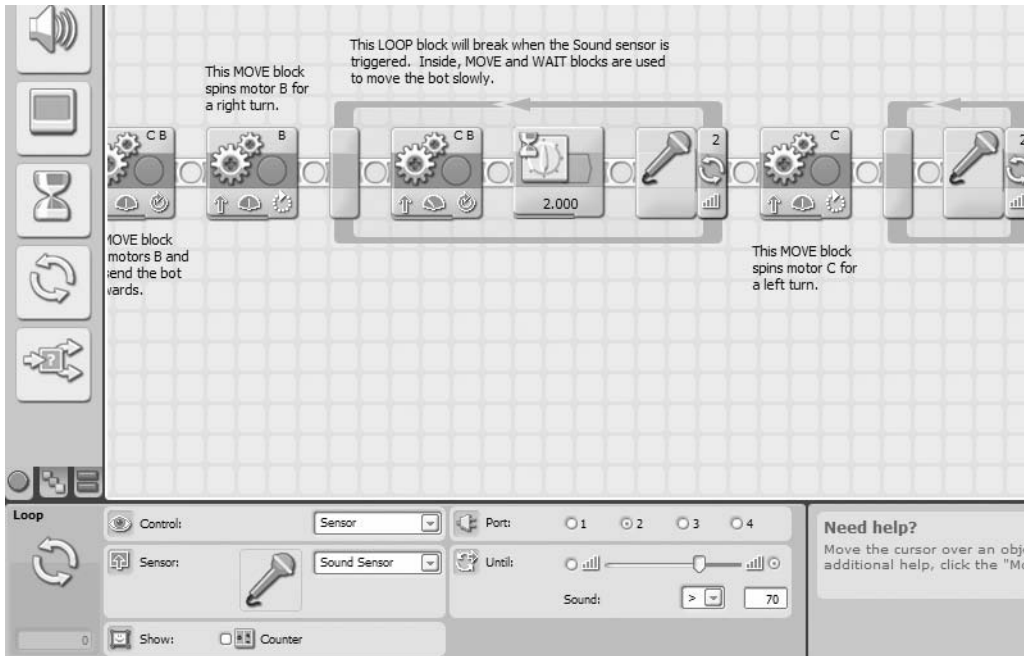


Figure 20-29. This LOOP block will break when the Sound Sensor is triggered.

A MOVE block is added inside the LOOP block to move the bot toward the figurine in quarter rotations (see Figure 20-30).

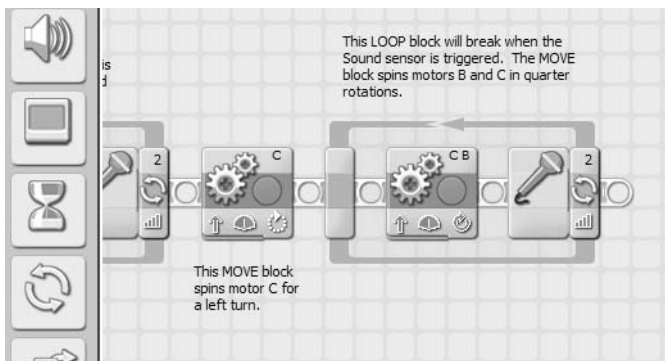


Figure 20-30. This MOVE block moves the bot toward the final figurine.

When the Sound Sensor is triggered, a MOVE block will close the cage on the final figurine (see Figure 20-31).

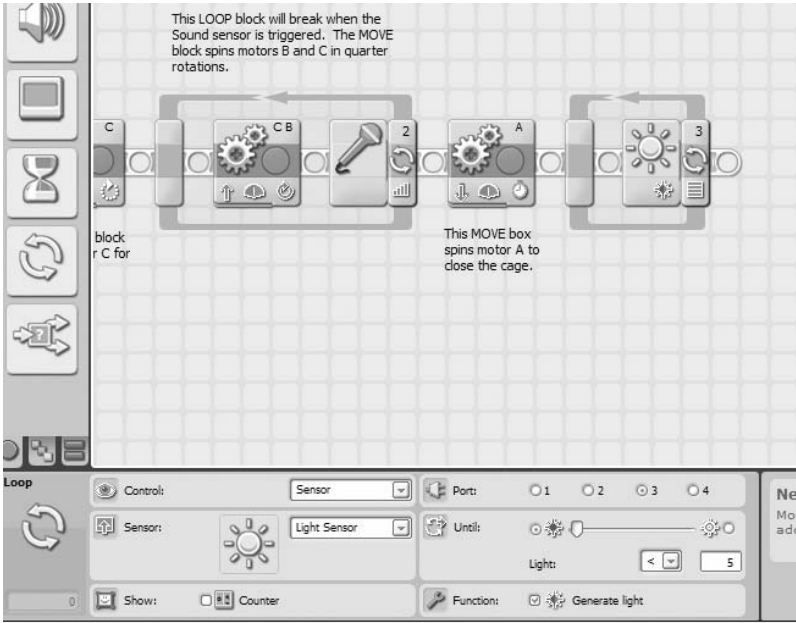


Figure 20-31. This MOVE block closes the cage on the final figurine.

I add in a final LOOP that will break when the Light Sensor is triggered by the black pressure plate on top of the sarcophagus (see Figure 20-32).

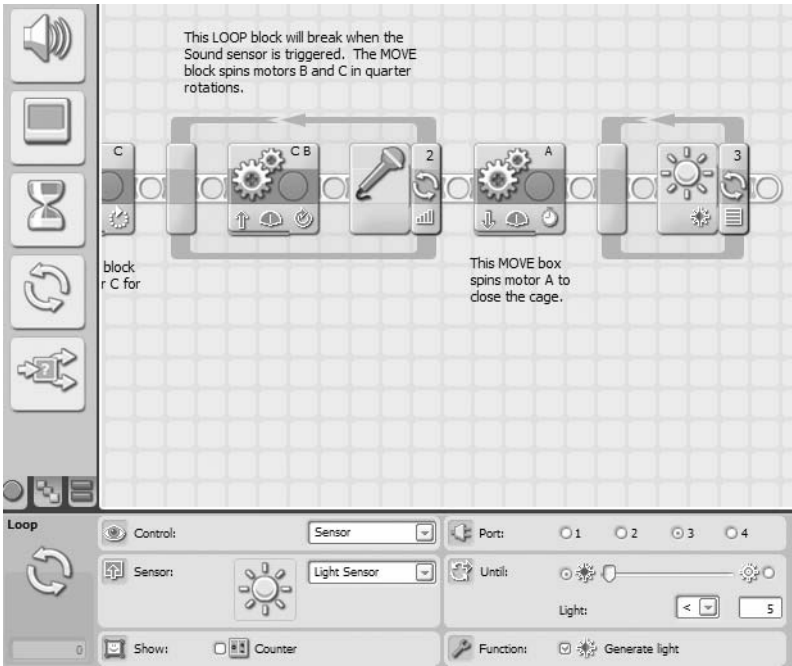


Figure 20-32. This final LOOP block breaks when the Light Sensor is triggered.

This last MOVE block will spin motors B and C in half rotations until the Light Sensor is triggered and the bot stops (see Figure 20-33).

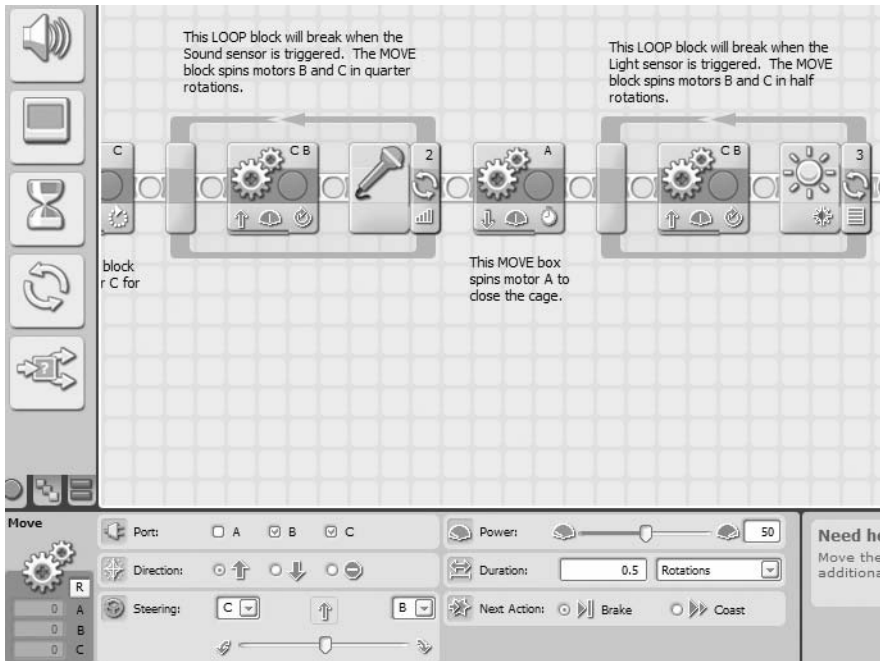


Figure 20-33. *This MOVE block stops spinning when the Light Sensor triggers.*

When the PushBot reaches the black pressure plate, the Light Sensor triggers the bot to stop moving. At this point, the final figurine should be resting on top of the sarcophagus.

If your robot's program succeeded, congratulations! If not, you'll need to test your bot quite a few times to make certain it works as expected. Set up your test environment and run as many as you can. When you're confident of your bot's abilities, run it one more time. If the bot accomplishes all the tasks, pat yourself on the back and get ready . . .

. . . It's time to investigate King Ixtua's burial chamber.

Story concludes in Chapter 21 . . .